

SPELL Project Plan Spring 2022

Avinash Persaud, Nandith Narayan, Carlos Cepeda
Advisor: Dr. William Shoaff

The Problem

Vocal synthesis is the process of creating artificial human speech. This can be achieved via machine learning. However, just like any other machine learning process, vocal synthesis requires a lot of data. This data requires labeling. Manually labeling this data takes up a large amount of time and effort.

Goal and Motivation

We aim to develop a standalone and user-friendly tool to assist users with labeling singing data for vocal synthesis and machine learning purposes.

The current tools have the following issues that we are trying to remediate:

- There is no one tool that can do it all, and this leads to developers needing to rely on multiple tools for labeling singing data
- The currently existing tools are also unintuitive and not user-friendly resulting in lots of time spent learning how to use the tool

Recap of Label Types

There are features to label in singing data. The most common of which are:

- Phonemes
 - Individual consonant or vowel sounds in a language that come together to form words
- Syllables
- Notes

Our application will allow the user to label the singing data with all of these layers.

Our Approach

- Allow the user to create mono labels graphically
 - Mono labels are a special label format that contains information about the data such as:
 - Start Time
 - End Time
 - Phoneme Identifier
- Ability to export the user generated label data in the HTS full label format
 - User will be able to export all their work with the click of a button
- Ability to automatically detect which phonemes are present in the audio
 - This helps reduce the amount of manual work the user has to do
- Ability to automatically align phonemes with the audio
 - This tool will detect where each phoneme must start and end, and place the phoneme accordingly, further reducing manual work on the user's part
- Ability to automatically save the users work given an interval of time between saves
 - This will ensure that the user will less likely lose the progress they have made

Our Approach (Cont.)

- Allow the user to export an existing project
 - This allows the user to transfer a project to another computer
- Allow the user to activate common tasks by using keyboard shortcuts
 - Labeling singing data requires the user to repeat simple tasks, and shortcuts can help alleviate the manual labor required.
 - These actions include:
 - Copying
 - Pasting
 - Duplicating
 - Moving
 - Etc.

Novel features

- Editing Shortcuts
 - Not found in common tools
 - Copy + Paste
 - Batch select and drag
- Integrated automatic phoneme detection+alignment
 - Multiple tools are usually required

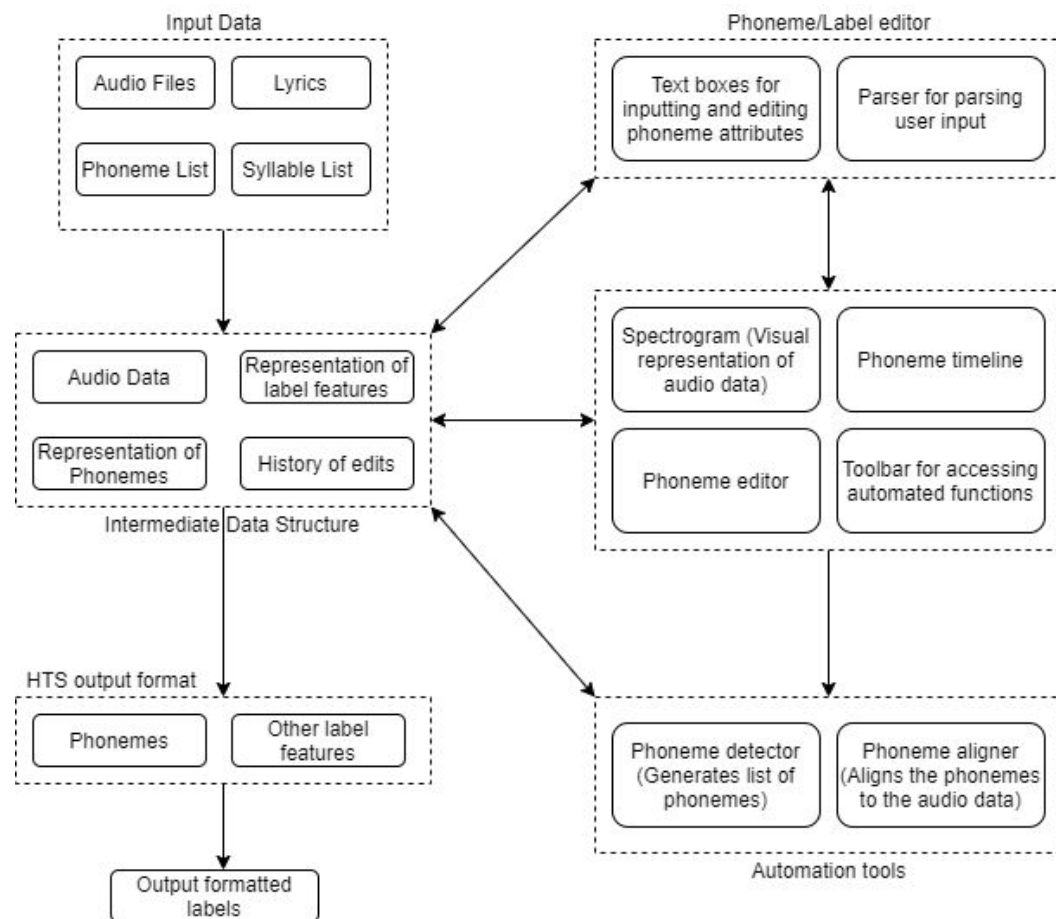
Technical Challenges

- Target output format
 - Formatting is information dense
- Automatic phoneme detection:
 - Phoneme Identification
 - Identify phoneme in a specified slice of time
 - Phoneme Boundary Detection
 - Detects transitions between phonemes

Technical Challenges (Cont.)

- Spectrogram generation speed
 - Slow
- Smooth dragging of UI elements
 - Mouse interaction

System Architecture Diagram

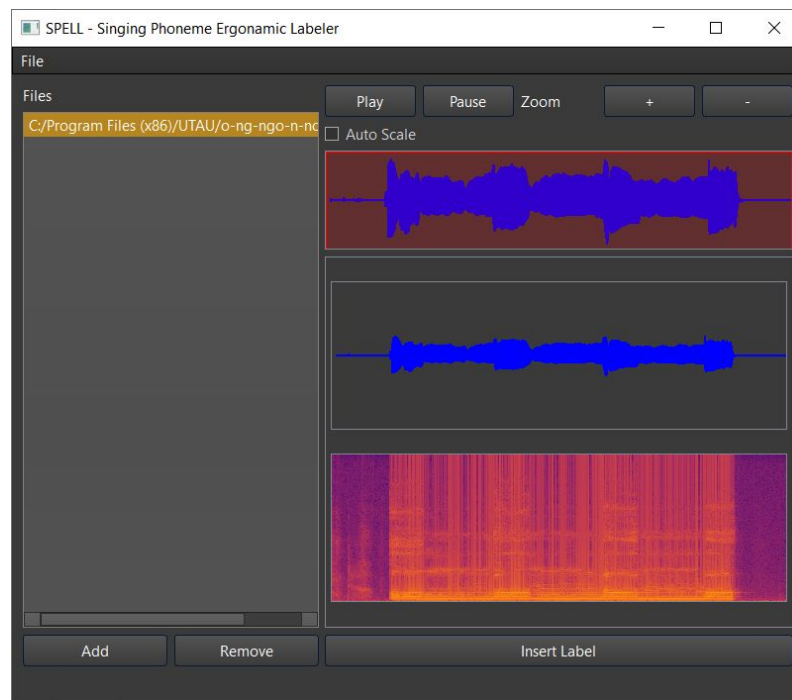


Evaluation

Success of our project will be measured using the following metrics:

- Fluidity of the UI
 - Get qualitative remarks on performance
 - Get a list of common tasks and how many clicks it takes to complete them
- Speed
 - Timing computationally expensive tasks such as spectrogram generation, loading a song, switching songs, and exporting data
- Accuracy
 - We will measure accuracy by comparing the auto generated phonemes against manual labels
- Reliability
 - We will measure this reliability by automating stress testing via a script to automatically add hundreds of songs at a time and perform actions like adding phonemes

Current UI layout



Milestone 4 (Feb 14)

- Basic phoneme feature input
- Basic project file IO
- Exporting labels to an output file
- Build test song database
- Create label output
- Add better navigation

Milestone 5 (Mar 21`)

- Complete timeline view
- Integrate phoneme boundary network
- Start note view
 - Piano roll
- Dictionary system

Milestone 6 (Apr 18)

- Implement, test, and demo which features/modules
 - File I/O
 - Phoneme labeling
 - Phoneme boundary detection
 - Output
 - Note Input
 - Lyric separation
- Test/demo of the entire system
- Evaluation results
- Create user/developer manual
- Create demo video

Milestone 4 Task Matrix

Task	Avinash	Nandith	Carlos
Complete intermediate data structure	50%	50%	0%
Create timeline feature widget	20%	20%	60%
Create project file format and I/O	20%	60%	20%
Create label output	50%	50%	0%
Build test song database	100%	0%	0%
Add better navigation	0%	80%	20%